REMARKS

Reconsideration and further examination of the application, as amended, is respectfully requested.

The second full paragraph of specification page 14 and the third full paragraph of specification page 21 were changed to include the application number for the referenced U.S. Patent application titled "A Protocol to Coordinate Network End Points to Measure Network Latency."

Description of the Present Invention

Applicants' invention is directed to a technique for accurately determining the latency of a selected path in a computer network. According to the inventive technique, a setup or signaling protocol is modified in a novel manner to establish a path reservation state at each intermediary node along the selected path. The path reservation state is associated with a given traffic flow having predefined parameters. Once the path setup process is complete, a source configures a test message in accordance with the predefined traffic flow parameters, time stamps the message, and transmits it to a receiver. As the message is routed through the network, it is identified at each intermediary node as matching the traffic flow, and thus is forwarded along the selected path. Upon receipt at the receiver, the test message is preferably returned to the source in a similar manner. By comparing the time at which the test message is returned with the time stamp contained in the message, an accurate latency of the selected path can be determined.

Significantly, by establishing the reservation state in advance of sending the test message, it can be assured that the test message will follow the selected path whose latency is to be measured without having to load any route information to the test message itself.

<u>§103</u>

At paragraph 2 of the Office action, claims 1-12 were rejected under 35 U.S.C. §103 as being unpatentable over Kompella et al., U.S. Patent No. 5,892,754 issued on April 6, 1999 (hereinafter "Kompella"), and Periasamy et al., U.S. Patent No. 6,023,733 issued on February 8, 2000 (hereinafter "Periasamy").

Claim 1 recites in part:

- ***"establishing a path state at each network node along the selected path for identifying a traffic flow having predefined parameters"***
- ***"in response to receiving the test message at each network node, forwarding the test message from the receiving network node to the next downstream network node along the selected path <u>by virtue of the previously established path states</u>"***
- ***"generating a path state setup message, the path state setup message for establishing a path state at one or more network nodes along a selected path of a computer network"***

Claim 8 recites in part:

- ***"inserting into the path state setup message a source routing option that lists one or more network nodes along the selected path"***
- ***"inserting into the path state setup message one or more parameters that define a selected traffic flow that is to be associated with a test message"***

Description of Cited References

Kompella teaches a flow control system for a computer network that is centered in user applications supplying data to the network (see Abstract). Upon request, the state of congestion in the computer network can be supplied to user applications (see Col. 2, lines 41-44). Specifically, a user application can specify upper and lower bounds of a Quality of Service (QoS) parameter of interest to the user application (see Col. 5, lines 55-59; Col. 6, lines 32-37; and, Col. 6, line 63 to Col. 7, line 9). A network parameter monitor sends event signals to the user application if the specified QoS parameter falls outside the identified upper or lower bound (see Col. 5, lines 60-67; and, Col. 7, line 19 to Col. 8, line 6). The user application then responds by modifying its flow of traffic into the network (see Col. 5, line 67 to Col. 6, line 2).

One of the QoS parameters mentioned by Kompella is latency. Kompella mentions that latency can be measured by computing the round trip delay of a test message, but does not provide any further details as to how this measurement may be performed (see Col. 7, lines 44-51).

Periasamy teaches a technique for representing the topology of a network as a tree structure where the root of the tree is a routing device (see Col. 4, lines 47-50). In the tree structure, tree nodes represent Local Area Networks (LANs) and arcs connecting the nodes represent other routing devices (see Col. 4, lines 50-51). Using the tree structure, a routing device may undertake different operations associated with the efficient routing of frames (see Col. 7, line 58 to Col. 8, line 18).

Differences between the Present Invention and the Cited References

Applicants respectfully urge that neither Kompella or Periasamy suggest or teach either individually or in combination Applicants' claimed steps of:

- ***"establishing a path state at each network node along the selected path for identifying a traffic flow having predefined parameters, and for forwarding messages matching the predefined parameters"***
- ***"in response to receiving the test message at each network node, forwarding the test message from the receiving network node to the next downstream network node along the selected path <u>by virtue of the previously established path states</u>"***

Kompella does not provide a teaching or suggestion for establishing a path state at each network node before issuing the test message nor for relying on the path state to correctly forward the test message along the desired path through the network. At best, Kompella suggests loading the test message itself with the path. This approach, however, severely limits Kompella to those computer networks that support messages which carry route information in their headers. By establishing path state in advance, applicant's invention works in networks where messages do not carry path information, thereby making the present invention usable in a much larger number of networks than Kompella.

At paragraph 3 of the Office action, claims 13 and 14 were rejected under 35 U.S.C. §103 as being unpatentable over Periasamy et al.

Claim 13 recites in part:

- ***"an options processor configured to implement one or more options included in a received path state setup message"***
- ***"a signaling protocol processor"***
- ***"wherein the options processor and signaling processor cooperate to implement a source routing option included in the path state setup message by initializing a path state associated with the traffic flow and forwarding the path state setup message to a next network node as identified in the source routing option"***

Neither Kompella nor Periasamy either alone or in combination teach or suggest an options processor and a signal protocol processor cooperating to initialize a path state in response to a path state setup message carrying one or more options. Rather, Periasamy describes a "route-discovery procedure" that distributes packets containing routing information to each station on every LAN in a network, and the stations, in turn, respond with packets containing MAC address and routing information to the source to enable the source to determine a path to the destination station. Further, with Periasamy as with Kompella, each packet carries the route it is to follow. Thus, there are no "path state setup messages" and no path states initialized in the network nodes along the path because the nodes rely on the routes specified in the packet to determine the packet's path. Indeed, the phrase "path state" does not appear in either Periasamy or Kompella.

For the reasons set forth above, applicants submit that all independent claims are believed to be in condition for allowance and that all dependent claims are believed to be dependent from allowable independent claims and therefore in condition for allowance.

Favorable action is respectfully solicited.

Please charge any additional fee occasioned by this paper to our Deposit Account No. 03-1237.

Respectfully submitted,

Michael J. Badzinski
Reg. No. 51,425
CESARI AND MCKENNA, LLP
88 Black Falcon Avenue
Boston, MA 02210-2414
(617) 951-2500

MARK-UP PAGES FOR THE SEPTEMBER 19, 2002, AMENDMENT TO U.S. PATENT APPLICATION SER. NO. 09/345,193

The replacement for the second full paragraph of page 14 resulted from the following changes:

To the extent source and destination ports are used by entities 202 and 204, the port numbers are preferably selected in accordance with commonly owned and co-pending U.S. Patent Application Ser. No. ~~[insert serial number]~~09/346,080 entitled A Protocol to Coordinate Network End Points to Measure Network Latency, which is hereby incorporated by reference in its entirety.

The replacement for the third full paragraph of page 21 resulted from the following changes:

Once the path states have been established within the devices along the selected path, source entity 202 preferably formulates and sends a test message to destination entity 204. In particular, latency determination engine 340 accesses time management facility 342 to create a time record or time stamp. Engine 340 places the time record into a test message and hands it down to the network communication facility 346 for transmission to destination entity 204. In the preferred embodiment, the format of the test message corresponds to the Network Endpoint Control Protocol (NECP), as described in previously referenced and incorporated U.S. Patent Application Ser. No. ~~[insert serial number]~~09/346,080. The network communication facility 346 preferably encapsulates the test message containing the time record in a corresponding packet. For example, the network communication facility 346 may first create one or more transport layer packets similar to the TCP packet of Fig. 1B, placing the test message from engine 340 into the data field 156. In the source port field 152, latency determination engine 340 directs communication facility 346 to load the value used in the source port field 460 of the sender template object 444 from the path state setup message 400 described above. In the destination port field 154, communication facility 346 is directed to load the value used in the destination port field 472 of the session object 446 from the path state setup message 400. The transport layer packet is then passed down to the respective

network layer where it may be encapsulated in a corresponding network layer packet, which, in the preferred embodiment, is preferably similar to IP packet 100 of Fig. 1A. Significantly, the test message utilized with the present invention does not include any options, thus there is no options area 130. In the IP SA field 126 of the test message, network communication facility 346 loads the IP address of source entity 202 (as utilized in the IP SA field 458 of the path state setup message 400), and, in the IP DA field 128, it loads the IP address of destination entity 204 (as utilized in the IP DA field 468 of the path state setup message 400). In the protocol field 122, communication facility 346 places the value, if any, previously utilized in the protocol field 470 from the path state setup message 400.